5780_Prelab_06

Shem Snow

March 22, 2024

1 : What is hysteresis and how does it help prevent bad behavior on digital inputs?

Hysteresis is when the threshold of a digital signal switching between high and low is changed. It's useful because signals are vulnerable to noise and if a digital signal is hanging out near its transition threshold then noise can cause the digital value to change. An example of applying Hysteresis is when a digital high signal is detected so the transition threshold is lowered so that the logic signal will not change its value as a consequence to noise.

2 : What is Quantization?

The lab says "Quantization is the process of mapping a high-resolution signal to a manageable lower-resolution one". In my own words, quantization is when you assign/map a continuous analog signal into a discrete one. We apply this in ADCs when we decide to use a finite amount of bits to represent all signals within a range. Those analog signals are quantized to as digital ones.

3 : What does Nyquist theory explain? What is the problem with sampling a signal too slowly?

Sampling too slowly allows for

1. Inaccuracy because amplitudes and frequencies can make great changes depending on your sampling rate.
2. Aliasing is when two signals of different frequencies happen to have the same value at every point that is sampled. When this happens, one signal transitions slower than the other. The consequence is that high frequency signals may be interpreted as low frequency signals and vice versa.

Nyquist theory explains the relationship between how often you sample a signal and how closely your quantized sample matches the actual signal. In order to represent a signal by sampling periodically, the sampling rate must be at least twice as fast as the original signal's frequency. Doing so will prevent aliasing.

4 : The maximum resolution of the ADC is 12-bits. How many quantization steps/values does this give us?

12 bits can represent $2^{12} = 4096$ unique analog values. This is pretty accurate for applications whose signals are low-to-medium magnitude.

5 : What are the steps to perform an ADC calibration?

1. Calibration can only be performed when the peripheral is stopped. The application must not use the ADC during calibration and must wait until it is complete. Calibration should be performed before starting A/D conversion. It removes the offset error which may vary from chip to chip due to process variation.
2. The calibration is initiated by software by setting ADCAL bit to 1. It can only be initiated when the ADC is disabled (when ADEN = 0). ADCAL bit stays at 1 during the whole calibration sequence.
3. It is then cleared by hardware as soon the calibration completes.
4. After this, the calibration factor can be read from the ADC_DR register (from bits 6 to 0).

The internal analog calibration is kept if the ADC is disabled (ADEN = 0). When the ADC operating conditions change (VDDA changes are the main contributor to ADC offset variations and temperature change to a lesser extend), it is recommended to re-run a calibration cycle. The calibration factor is lost each time power is removed from the ADC (for example when the product enters Standby mode).

6 : What's the difference between right and left-aligned data in the DAC registers?

The left and right aligned modes are used for selecting whether the 16-bit number digital value is stored in the upper (left) or lower (right) bits of the DAC data register. In the case of left-aligned mode, precision of the lower 8 bits is lost.

7 : What DAC register would you use to write 8-bit right-aligned data? (use the peripheral reference manual)

One of the DAC_DHR8Rx (x = 1, 2) data holding registers because they are 8-bit aligned (the peripheral manual says so).

8 : Name something you found confusing or unclear in the lab manual. If everything was clear, simply answer that you didn't have any issues.

I'm still a little confused about how the left and right aligned modes make use of only part of a data register. I don't see the advantage of writing to only one part. Also, the data holding register that is left- aligned uses bits [15:4]. I thought the idea was to only use half of the register and therefore only 8 bits.